専攻分野	工学
専攻区分	情報通信工学

平成 25 年度情報通信工学科卒業研究

テーマ名： 残留塩素自動測定システムの開発に関する研究

氏　　名： 長川 裕哉

指導教員： 真鍋 克也

目 次

第 1 章　概要 1

第 2 章　システム構成 2

2.1　塩素制御装置のシステム構成 3

2.2　システムの全体構成 4

第 3 章　ポータブル残留塩素計の取り替え 5

第 4 章　Perl を用いたプログラム 6

4.1　プログラムのフローチャート 6

4.2　シリアルパラメータの詳細設定 7

4.3　ファイルへの書き込み 8

第 5 章　計測テスト 9

5.1　タスクの設定 10

第 6 章　グラフ描画 11

6.1　RRDtool でのグラフ描画 11

6.2　RRDtool の利用方法 11

6.3　ラウンドロビンデータベースの更新 12

6.4　Cron の設定状況 13

6.5　グラフの表示 14

第 7 章　今年度のシステム構成 15

第 8 章　むすび 16

第 9 章　今後の課題 17

第 1 章　概要

屋外プールでは残留塩素濃度が，その日の天候や気温によって激しく変化し，既定値を外れると現場の装置まで足を運び，塩素注入装置のスイッチを切り替えなければならない．本校ではプールの残留塩素濃度測定のポータブル残留塩素濃度測定器が用いられており，遠隔制御によって残留塩素濃度を管理している．前年度までの卒業研究で，新しい無線 LAN シリアルアダプタの設定，Samba，Web サーバの移行，計測コンピュータの移行，計測における機能の追加，塩素注入装置のプログラムの追加を行った．

今年は，VisualBasic2010 言語を用いた測定プログラムを perl 言語を用いたプログラムに書き換え，ポータブル残留塩素計の取り替え，ポータブル残留塩素計と新しい無線 LAN シリアルアダプタとの接続の確認，タスクの設定，計測テスト，RRDtool でのグラフ描画を行った．これにより，プールの残留塩素濃度の遠隔制御システムの実用的，日常的な管理，運用を行うことができた．その中で問題の発見し，改善していった．

第 2 章　システム構成

昨年度まで実稼働していたシステムの構成を図 1 で表す [1,2].

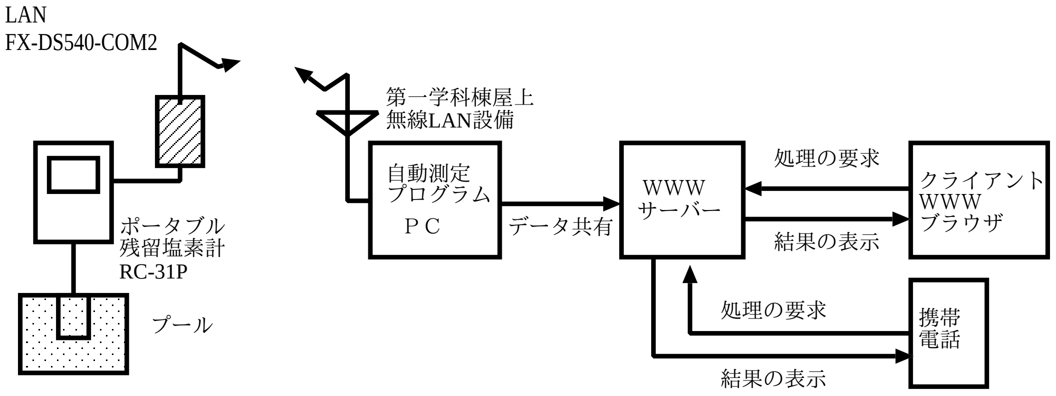

図 1　昨年度の自動測定システム

昨年度同様構築したシステムに使用した PC などの機能等を示す. サーバー：DELL Dimension 9150

サーバー OS：Sun Solaris Express

グラフ作成ツール：RRDtool

残留塩素測定器：東亜 DKK RC-31P

自動測定システム稼働 PC：Scenage ValueT335DUK24/30

2.1　塩素制御装置のシステム構成

塩素制御装置のシステム構成を図 2 に示す．本研究では，はじめに昨年度同様 AC 電源を使い測定を行う. 今年度より，プールに新しく設置された AC 電源に切り替え，昨年度までの AC 電源は詫間用，今年度に使う AC 電源は高松用にし，AC 電源から新しく購入した残留塩素計の AC アダプタを介して配線ケーブルをつなぎ測定を行うことで誤差がなくなった．また，ポータブル残留塩素計昨年度同様 RC-31P を 2 つ使用している．

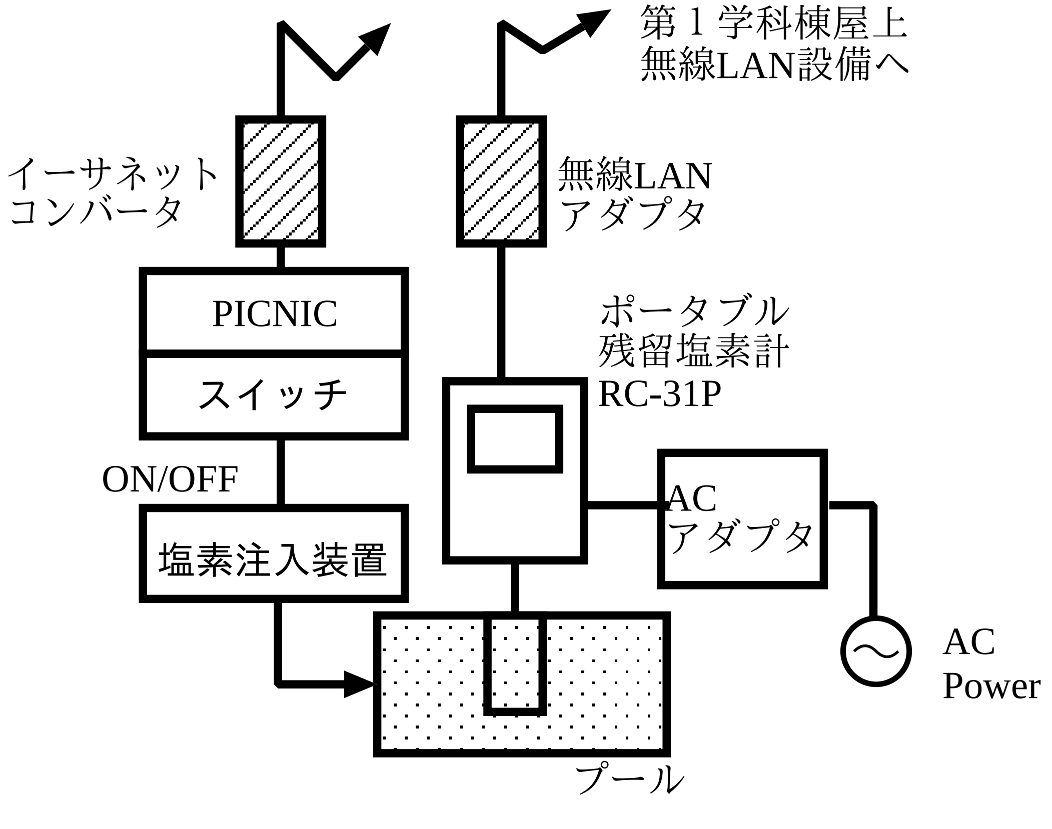

図 2　塩素制御装置のシステム構成図

2.2　システムの全体構成

システムの全体構成図を図 3 に示す．イーサネットコンバータは，本体有線 LAN ケーブルでルータへ接続する必要がある端末を無線化できる装置である．PICNIC は，周辺機器接続用のインターフェイスを用意した PIC マイコンにネットワークインターフェイスを持っているものである．リレーは，PICNIC に接続した塩素注入装置を制御するものである．塩素注入装置は，塩素を注入するための装置である．無線 LAN アダプタは，PC に無線 LAN への接続機能を追加するための拡張機器である．ポータブル残留塩素計は，無試薬で水道水やプール水などの残留塩素を測定できるものである．自動測定プログラム PC は，計測用 PC のことである計測用プログラムを動作させる WindowsPC．自動測定プログラムにより，残留塩素計で測定する．現在，過去のデータをクライアント WWW ブラウザで確認し，WWW サーバを使用して無線 LAN により塩素注入装置を AUTO/OFF 切り替えできる [1,2].

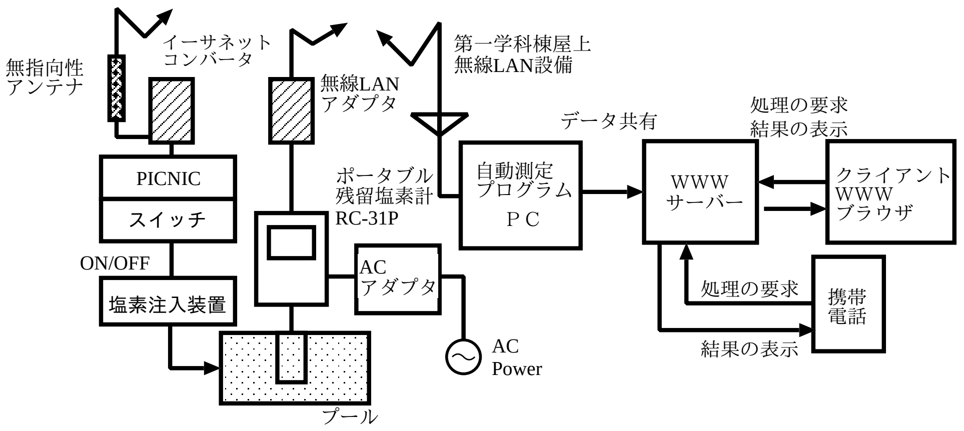

図 3　一連のシステムの全体構成図

第 3 章　ポータブル残留塩素計の取り替え

昨年まで使用していたポータブル残留塩素計は (RC-31P) 詑間に．新しいポータブル残留塩素計 (RC-31P) は高松で使うため購入した．新しいポータブル残留塩素計は 10 月の実稼働以降に取り替えを行った．図 4 にポータブル残留塩素計 (RC-31P) を示す．

図 4　ポータブル残留塩素計 (RC-31P)

第 4 章　Perlを用いたプログラム

今年度より，perl 言語を用いた測定プログラムに一から書き換えを行った．今年から本校の Windows XP が使用できなくなるためパソコンを移行する必要があった．実際に得られた結果は後半に示している第 6 章にグラフによる観測結果，付録に Perl のプログラムを記述してある．

4.1　プログラムのフローチャート

perl 言語を用いたプログラムのフローチャートを図 5 で示す．シリアルポートを Open したのち 4 回のループを繰り返す．シリアルポートに要求コードを書き込む．値が得られればループを抜けて，ファイルに残留塩素濃度，水温，日時を書き込む．

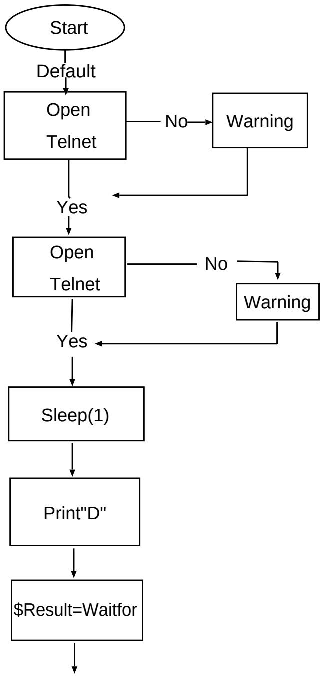

図 5　プログラムのフローチャート

4.2　シリアルパラメータの詳細設定

ポータブル残留塩素計と無線 LAN 通信するために必要なパラメータを設定した．以下に設定内容を示す [1].

FXR2000 シリアル設定

ボーレート　: 19200(bps)

データビット　: 8 (bit)

パリティ　: なし

ストップ　: 1 (bit)

ソフトウェアフロー制御 : なし

ハードウェアフロー　: なし

受信バッファサイズ　: 1024

FXR2000 TCP 設定

IP アドレス　: 192.168.201.8

サブネット　: 255.255.255.0

ゲートウェイ : 192.168.201.254

詳細設定 (2)

FXR2000 TCP コネクション

TCP シリアル機能　: 有効

動作モード　: サーバ

TCP ポート番号　: 8000

送信タイムアウト (ミリ秒) : 無効

アイドルタイムアウト (秒) : 無効

FXR2000 詳細設定

VAP　: 変更なし

SNMP　: 変更なし

ネットワークタイム機能 : 有効

NTP サーバ　: 150.15.8.33

4.3　ファイルへの書き込み

測定結果をファイルに自動で書き込み記録していく．ネットワークがダウンした時，Samba サーバがダウンした時，Samba のパスワードを入力し忘れていた時の可能性を考慮しローカルホストにも記録している．ローカルホストでは，正常時とエラー時にファイルを分けて記録している．

- Samba サーバ

ファイル名：log.dat

アドレス：\\150.15.73.90\t09028\public_html\web_tool

- ローカル

ファイル名：log.txt

アドレス：H:\Documents and Setting\GuestUser\My Documents\t09028

ファイル名：errlog.txt

アドレス：H:\Documents and Setting\GuestUser\My Documents\t09028

第 5 章　計測テスト

ピング試験でプールとの無線 LAN 接続を確認したのち，ポータブル残留塩素計 (RC-31P) と perl を用いた計測プログラムのテストを行った．プールの使用は 10 月までだったため，プールサイドで 1 週間計測した．半透明のボックスの中には無線 LAN シリアルアダプタ，電源，ポータブル残留塩素計が入っている．図 6 に計測テストの様子を示す．

図 6　計測テスト

5.1　タスクの設定

プログラムを連続稼働するためタスクの設定を行った．日照による水温の変化を考え 8:00 から 18:00 まで，18:00 から 8:00 までタスクを 2 回にわけて設定した．以下にタスクの設定を示す.

8:00 から 18:00 まで

- 継続期間 → 10 時間
- 間隔 (日単位) → 毎日
- 間隔 (分単位)→ 5 分

18:00 から 8:00 まで

- 継続期間 → 14 時間
- 間隔 (日単位) → 毎日
- 間隔 (分単位)→ 5 分

第 6 章　グラフ描画

6.1　RRDtool でのグラフ描画

前年度の構築された RRDtool のデータベースに今年度もグラフを描画した．RRDtool とは，収集したデータの格納にラウンドロビンデータベースを利用し，その格納したデータを利用してグラフ描画を行うものである．ラウンドロビンデータベースとはあらかじめあつかうデータ数が決められれていて，新しいデータを追加する際には古いデータを順次上書きして保存するようなデータベースである [3].

6.2　RRDtool の利用方法

RRDtool でグラフを作成する場合には以下の 3 つの作業が必要である．

1. ラウンドロビンデータベースの作成
2. ラウンドロビンデータベースの更新
3. グラフ描画

これらの作業にはすべてコマンドを使用する．データの器となるラウンドロビンデータベースを定義しておく必要がある．前章で行った計測テストの結果をグラフに描画するためラウンドロビンデータベースの更新のプログラムの書き換えを行った．

6.3　ラウンドロビンデータベースの更新

前年度作成されたラウンドロビンデータベースのアップデートを行うためのプログラムの書き換えを行った．温度，塩素濃度の更新：misaki.pl

```
#! /bin/perl

# 変数定義
$ENV{"RRD_DEFAULT_FONT"}=
    "/usr/openwin/lib/locale/ja/X11/fonts/TT/HG-GothicB.ttf";
$ENV{"LANG"}="ja_JP.UTF-8";
$RRD="/home/t09028/public_html/enso.rrd";          # RRD ファイルのパス
$PATH="/usr/local/rrdtool-1.2.19/bin";      # rrdtool コマンドの場所指定
$COMMUNITY="community_name";            # SNMP のコミュニティ名指定
$REMOTE="remote_host";                  # 監視対象のホスト名 orIP アドレス
chdir("/home/t09028/public_html");
$In=‘/bin/tail -1  /home/t09028/public_html/web_tool/takamatsu.dat)
($hizuke,$zikoku,$ondo,$noudo)=split(’ ’,$In);
$ondo=~ s/ //g;
$noudo=~ s/ //g;
# ラウンドロビンデータベースの更新
system("$PATH/rrdtool update $RRD N:$ondo:$noudo");
open(LOG,">/home/t09028/public_html/rrd_log.dat");
print LOG "updated! $zikoku $ondo $noudo\n";
close(LOG);
```

6.4　Cron の設定状況

プールの運用は 10 月までだが計測テストをするために cron の設定に 1 月も稼働するようプログラムに追加を行った．cron は 1 行目が描画と更新，2 行目が残留塩素のチェックを示している．行の 1 列目には分，2 列目に時，3 列目に日，4 列目に月，5 列目に曜日を示している．*は全てにマッチしており時ならば 24 時間を表している．cron の編集は，putty より 7hou.cn.kagawa-nct.ac.jp へログイン後 crontab -e でできる．putty とは，windows において SSH を使ったサーバへ安全にリモートログインを行うためのソフトウェアである．

```
3,8,13,18,23,28,33,38,43,48,53,58 * * 6,7,8,9,10,11,12,1,2 * \
 /home/t09028/public_html/misaki.pl >/dev/null
3,13,23,33,43,53 * * 6,7,8,9,10,11,1,2 * \
 /home/t09028/public_html/takuma.pl >/dev/null
```

6.5　グラフの表示

以上のことより図7のようなRRDtoolを用いたグラフが完成した．横軸が時間，縦軸が塩素濃度または温度を表しており，上から一日の塩素濃度，一日の水温，一週間の水温を表している．プール運用期間に測定ができなかったので塩素の値が低いまま表示された．

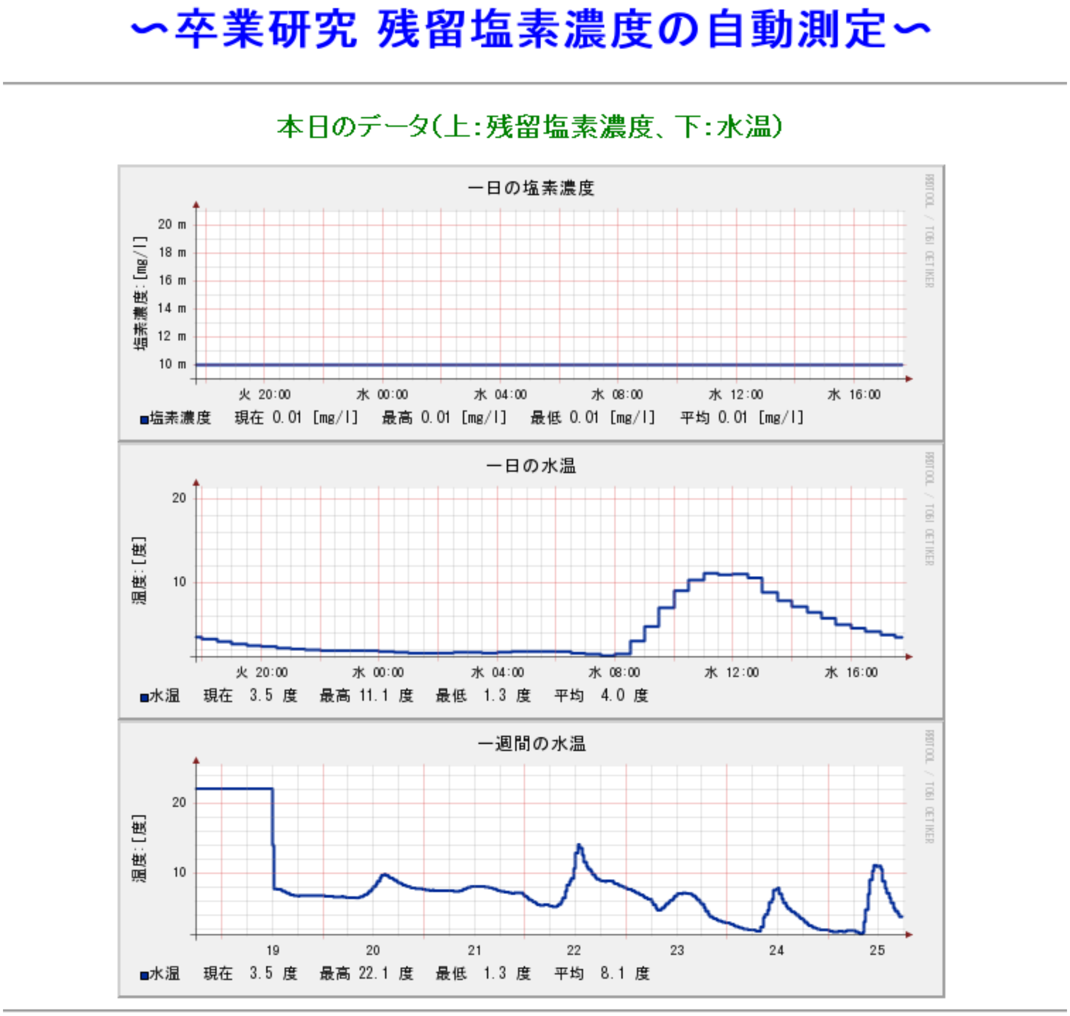

図7　RRDtoolによるグラフ

第 7 章　今年度のシステム構成

今年度開発したシステムの構成を図 8 に示す．無線 LAN シリアルアダプタを FX-DS540 から FXR2000 に変更した．

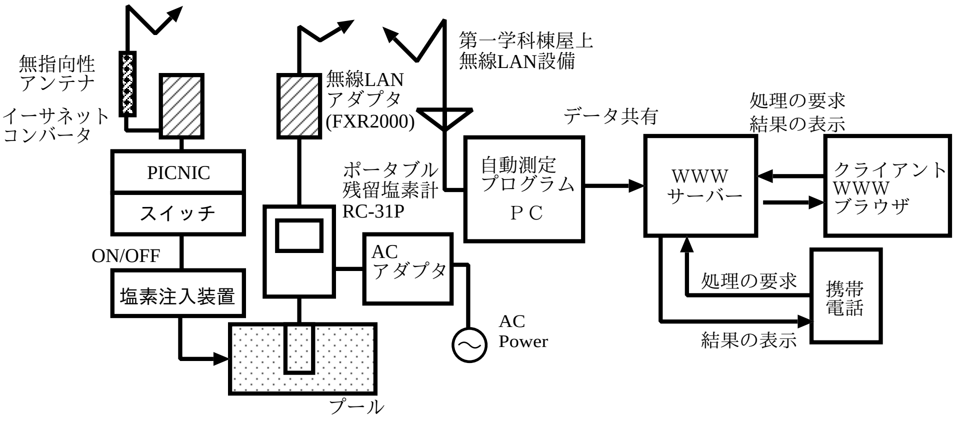

図 8　今年度の自動測定システム

第 8 章　むすび

今回の研究により，新しく購入したポータブル残留塩素計の AC アダプタを AC 電源と介して接続することで正しい測定値が得られ，電力源として AC 電源が利用可能になった．また，計測プログラムを Visual Basic 2010 言語から perl 言語に書き換えを行い，Samba サーバ，ローカルと 2 つのファイルへ記録することができた．計測用の PC も新しい計測 PC にタスクを設定し実際に，perl 言語を用いたプログラムの計測テストを 1 週間行いプログラムの継続稼働，ファイルへの書き込み，日時や水温と塩素濃度の測定値などを確認した．また cron に計測期間の 1 月稼働を追加することで，昨年度作成された RRDtool のグラフに計測結果を表示し Web 上でも計測テストの結果を確認することができた.

以上のことより，プールの管理システムおよび測定を新システムに更新できた.

第 9 章　今後の課題

今回の研究はここまでに留まったが，次の 3 つの点が今後の課題だと考えらる．

- プログラムの実稼働

今年度構築した言語のプログラムを実際にプール運用期間中に試す必要がある．今現在の段階では遠隔で水温や塩素濃度を計測することしかできない．今後自動で調整できるプログラムを開発することが望まれる。

- プログラムを公開すること

プログラムを汎用性の高いものに書き換えてから公開する必要がある．現在の状態では計測用のプログラム自体を書き換えなければ他校では利用できないが，汎用性の高いものであれば設定ファイルを書き換えるだけで利用できる．よって，プログラムを公開する事により，他校のプールでも同じようにシステムが利用できると思われる．

謝辞

最後に，本研究にあたり全面的に御指導を頂いた真鍋克也先生に厚く御礼申し上げます.

参考文献

[1] 持主 忠：プールの残留塩素濃度計測システムの開発に関する研究：平成 21 年度 香川高等専門学校情報通信工学科 卒業論文

[2] 三崎 友樹：プールの残留塩素濃度自動計測システムに関する研究：平成 22 年度 香川高等専門学校情報通信工学科 卒業論文

[3] 阿川 翔平：電波式変位計における気象による影響観測システムの構築に関する研究：平成 19 年度 香川高等専門学校情報通信工学科 卒業論文

[4] 白井 千智：VB2010 による残留塩素測定システムの開発に関する研究：平成 23 年度 香川高等専門学校情報通信工学科 卒業論文

[5] ポータブル残留塩素計 RC-31P 取扱説明書，東亜 DKK 株式会社

[6] 生田 誠三：LATEX 2ε 文典，朝倉書店

付録

付録として perl 言語を用いた残留塩素測定プログラムの全文を以下に記す．まず始めに詫間用のプログラムである.

```
#!/bin/perl
#use strict;
$host="192.168.201.8";
$port=8000;
$Input_File="newfile";
$bin=0;
use Net::Telnet;
#open(STDERR,">&STDOUT");
$|=1;
#print "password=",$password,"\n";
#print "csvfile=",$csvfile,"\n";
$com=0;
my $tn = new Net::Telnet(Host => $host, Port => $port, Timeout => 40 ,
 Input_log => $Input_File, Binmode => $bin)
    or warn print"connect failed: $!";
#print "stage1 Telnet open\n";
$tn->open ($host) or warn print"connect failed: $!";
#print "stage2 ",$tn,"\n";
sleep(1);
my $result;
#    ($result) = $tn->waitfor('/Connected to /i');
#    (@result1) = $tn->print($pass);
#print @result1;
#print 'Hostname: ',"\n";
#    ($result) = $tn->waitfor('/Command/i');
```

```
$r=0;
for (1..4) {
    $result = "";
#    print "stage3 send command D\n";
#    $result = $tn->print("\n");
#    print "stage4 ",$result,"\n";
#    $result = "";
#    $result = $tn->waitfor('/,ER/i');
#    print "stage5 ",$result,"\n";
    $result = $tn->print("D");
#    print "stage4 ",$result,"\n";
    $result = $tn->waitfor('/D/i');
#    print "stage5 ",$result,"\n";
    $r=1 if ($result =~ /^D/);
    last if ($r);
}
$tn->close or die "close fail: $!";
open(IN,$Input_File);
while (<IN>) {
    ($d,$a,$t,$c) = split(/,/,$_);
    last if (5 <$t && $t < 40);
}
close(IN);
$com =1 if ($result =~ /^D/);
$date = `date +%Y%m%d`;
chop($date);
$time = `date +%H%M%S`;
chop($time);
open(OUT,">>takuma.dat");
```

```
#printf "%s %s %4.1f %4.2f\n",$date,$time,$t,$c;
printf OUT "%s %s %4.1f %4.2f\n",$date,$time,$t,$c;
close(OUT);
```

次に高松用のプログラムである.

```
#!c:\perl64\bin\perl
#use strict;
$host="150.15.72.2";
$port=8000;
$Input_File='C:\Users\塩素測定システム\Documents\ensoperl\newfile';
$bin=0;
chdir("C:\Users\塩素測定システム\Documents\ensoperl");
use Net::Telnet;
#open(STDERR,">&STDOUT");
#bin(STDERR);
$|=1;
#print "password=",$password,"\n";
#print "csvfile=",$csvfile,"\n";
$com=0;
my $tn = new Net::Telnet(Host => $host, Port => $port, Timeout => 40 ,
 Input_log => $Input_File, Binmode => $bin)
    or warn print"connect failed: $!";
#print "stage1 Telnet open\n";
$tn->open ($host) or warn print"connect failed: $!";
#print "stage2 ",$tn,"\n";
sleep(1);
my $result;
#    ($result) = $tn->waitfor('/Connected to /i');
#    (@result1) = $tn->print($pass);
```

```
#print @result1;
#print 'Hostname: ',"\n";
#    ($result) = $tn->waitfor('/Command/i');
$r=0;
for (1..4) {
    $result = "";
#    print "stage3 send command D\n";
#    $result = $tn->print("\n");
#    print "stage4 ",$result,"\n";
#    $result = "";
#    $result = $tn->waitfor('/,ER/i');
#    print "stage5 ",$result,"\n";
    $result = $tn->print("D");
#    print "stage4 ",$result,"\n";
    $result = $tn->waitfor('/D/i');
#    print "stage5 ",$result,"\n";
    $r=1 if ($result =~ /^D/);
    last if ($r);
}
$tn->close or die "close fail: $!";
open(IN,$Input_File);
#print "stage6 newfile open\n";
while (<IN>) {
    ($d,$a,$t,$c) = split(/,/,$_);
    last if (5 <$t && $t < 40);
}
#print "stage7 newfile read!\n";
close(IN);
($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) = localtime(time);
```

```
$year += 1900;
$mon += 1;
#print "$year 年$mon 月$mday 日 ($youbi[$wday]) $hour 時$min 分$sec 秒\n";
#$date = ‘date +%Y%m%d‘;
$date = sprintf("%4d%02d%02d",$year,$mon,$mday);
#chop($date);
#$time = ‘date +%H%M%S‘;#!c:\perl64\bin\perl
#$time = ‘date +%H%M%S‘;
#chop($time);
$time = sprintf("%02d%02d%02d",$hour,$min,$sec);
open(OUT,’>> C:\Users\塩素測定システム\Documents\ensoperl\takamatsu.dat’);
printf "%s %s %4.1f %4.2f\n",$date,$time,$t,$c;
printf OUT "%s %s %4.1f %4.2f\n",$date,$time,$t,$c;
close(OUT);
sleep(1);
print(’C:\windows\system32\ftp \
     -s:\Users\塩素測定システム\Documents\ensoperl\command.txt\
     >\Users\塩素測定システム\Documents\ensoperl\ftp.log’,"\n");
system(’C:\windows\system32\ftp \
     -s:\Users\塩素測定システム\Documents\ensoperl\command.txt\
     >\Users\塩素測定システム\Documents\ensoperl\ftp.log 2>&1’);
```